

```
REPORT ZIMPORTING_EXPORTING.
DATA : ST TYPE STRING VALUE 'vikas SHRivastAVA', RES TYPE string.
```

```
DATA O1 TYPE REF TO ZSTRING_OPERATIONS.
CREATE OBJECT O1.
```

```
*CALL METHOD O1->TO_UPPER
* EXPORTING
*   STR1  = ST
* IMPORTING
*   STR2  = RES
*   .
```

```
" OR CALL DIRECTLY ....
```

```
O1->TO_UPPER( EXPORTING STR1 = ST IMPORTING STR2 = RES ).
WRITE RES.
```

```
*****"++++++
+"*****
```

```
DATA STLEN TYPE I.
```

```
*CALL METHOD O1->GET_LENGTH
* EXPORTING
*   STR  = ST
* RECEIVING
*   LEN  = STLEN
*   .
```

```
"OR
```

```
STLEN = O1->GET_LENGTH( STR  = ST ).
```

```
WRITE / STLEN.
```

```
*****"++++++
+"*****
```

```
ST = 'HI / HELLO / HOW  / hi ./ / ARE  / YOU / DUDE . '.
```

```
*CALL METHOD O1->CONDENSE_STR
* CHANGING
*   STR  = ST
*   .
```

```
"OR
```

O1->CONDENSE_STR(CHANGING STR = ST).

WRITE / ST.

WRITE /.

```
+++++
+

```

DATA : S1 TYPE STRING, S2 TYPE STRING,S3 TYPE STRING, S4 TYPE STRING.

*CALL METHOD O1->SPLIT_STR

* EXPORTING

* MAIN_STR = ST

* IMPORTING

* STR1 = S1

* STR2 = S2

* STR3 = S3

* STR4 = S4

* .

O1->SPLIT_STR(EXPORTING MAIN_STR = ST IMPORTING STR1 = S1 STR2 = S2
STR3 = S3 STR4 = S4).

WRITE : / 'AFTER SPLIT :', / , 'S1 = ', S1 COLOR 7, / , 'S2 = ', S2 COLOR 3, / , 'S3 = ',
S3 COLOR 4, / , 'S4 = ', S4 COLOR 5.

```
+++++
+

```

*CALL METHOD O1->CONCAT_STR

* EXPORTING

* STR1 = s1

* STR2 = s2

* STR3 = s3

* STR4 = s4

* IMPORTING

* MAIN_STR = st

* .

" or

o1->concat_str(EXPORTING str1 = s1 str2 = s2 str3 = s3 str4 = s4 IMPORTING
main_str = st).

SKIP 2.

WRITE : 'After concatenating' COLOR 1.

SKIP 1.

write :/ st.

DATA : old type string VALUE 'hi', new TYPE string VALUE 'hello'.

```
*CALL METHOD O1->REPLACE_STR
* EXPORTING
*   OLD_STR = old
*   NEW_STR = new
* CHANGING
*   MAIN_STR = st
*   .
```

o1->replace_str(EXPORTING old_str = old new_str = new CHANGING main_str =
st).

skip 2 .

write :/ 'after concatenation ' , /, st COLOR 2.